今天我们来练习如何创建按钮的监听和使用菜单栏功能。下面是两张微信的截图，你曾否想过在这个界面上的这两个功能是怎么实现的？

|  |  |
| --- | --- |
| Macintosh HD:Users:liuk:Downloads:IMG_1371.PNG | Macintosh HD:Users:liuk:Downloads:IMG_1370.PNG |

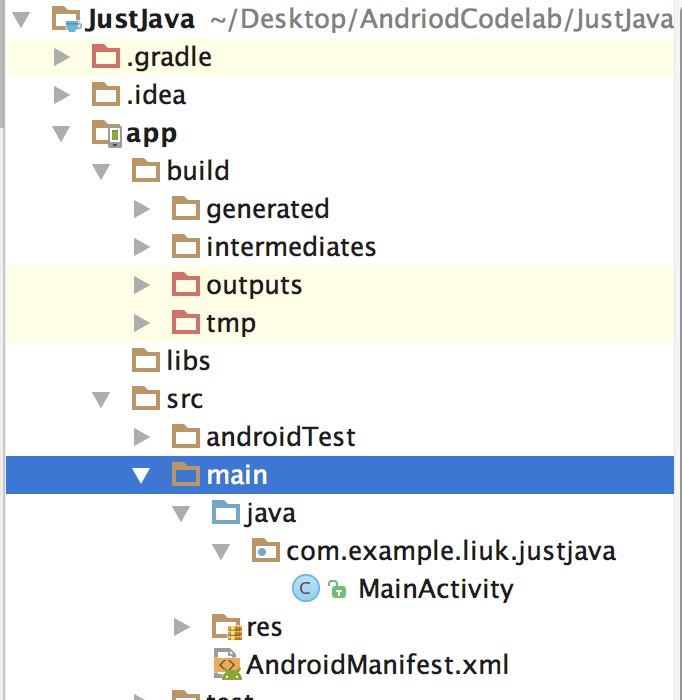
接下来大家跟我来一步一步实现按钮的单击监听和长按监听。

1. 首先，像往常一样创建一个空白的项目。
2. 然后，将我所给的xml文件复制到你layout文件当中。文件名activity\_main.xml。

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/activity\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context="com.example.liuk.justjava.MainActivity"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="长按下面的二维码识别"  
 android:textSize="16dp"  
 android:textColor="@android:color/background\_dark"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginLeft="8dp"  
 android:id="@+id/textview\_barcoderesult"**/>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="0"  
 android:textSize="16dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginLeft="8dp"  
 android:id="@+id/textView\_numberOfOrder"**/>  
  
 <**Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="点击我"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginRight="8dp"  
 android:background="@android:color/holo\_blue\_light"  
 android:id="@+id/button\_order"**/>  
  
 <**Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginRight="8dp"  
 android:text="长按我"  
 android:background="@android:color/holo\_blue\_dark"  
 android:id="@+id/button\_toast"** />  
  
 <**ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:id="@+id/imageview\_barcode"**/>  
 *<!--android:src="@drawable/barcode\_test"-->*</**LinearLayout**>

1. 在MainActivity的onCreate方法中加入以下的语句，文件目录如下：

app/src/main/java/MainActivity



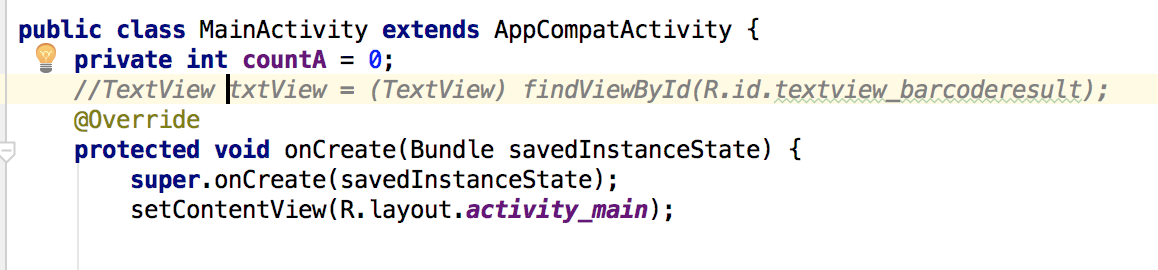
文件step1

**final** Button button = (Button) findViewById(R.id.***button\_order***);  
**final** Button buttonToast = (Button)findViewById(R.id.***button\_toast***);

button.setOnClickListener(**new** View.OnClickListener(){  
  
 @Override  
 **public void** onClick(View v){  
 TextView numberOfOrders = (TextView) findViewById(R.id.***textView\_numberOfOrder***);  
 numberOfOrders.setText(**""**+(++**countA**));  
//这个地方是用来写当检测到点击事件的时候会执行的语句。  
 }  
 });  
  
buttonToast.setOnLongClickListener(**new** View.OnLongClickListener(){  
 @Override  
 **public boolean** onLongClick(View v) {  
 Toast.*makeText*(MainActivity.**this**,**"这是一个长按事件！"**,Toast.***LENGTH\_LONG***).show();  
//这个地方是用来写当检测到长点击事件的时候会执行的语句。在这个方法中你可以写任意合法的语句，

//但是其他的地方是固定的，好了再回到刚才你阅读的地方。  
 **return false**;  
 }  
});

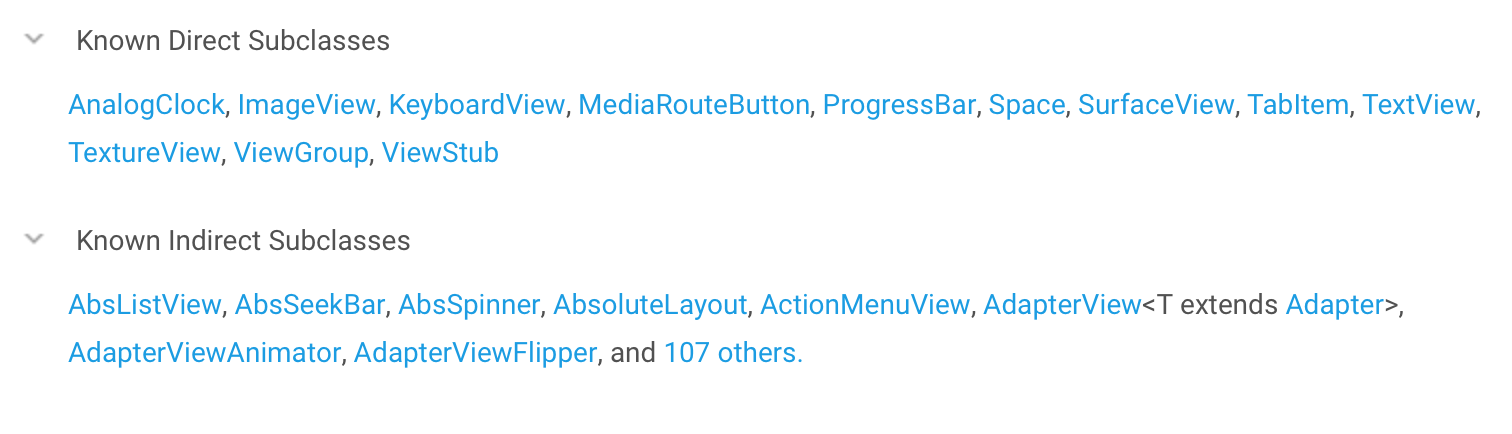
这个时候会出现错误，因为我们没有定义countA这个变量，需要你在下面图示位置加入变量：**private int countA** = 0;



现在就已经完成对按钮的监听拉，运行你的app，点击“点击我按钮”计数器会加一，而长按“长按我”这个按钮，会在app底部出现提示“这是一个长按事件”，这其实和微信那个长按按钮编辑家纯文本是一样的。

这个时候我们回到代码部分，看看我写的注释。

1. 其实不只有button可以设定监听，所有继承自View的类都可以，就像你遗传自你父亲母亲的那些基因，只要是显性的都可以使用。所以我们来看看有哪些View可以调用监听？

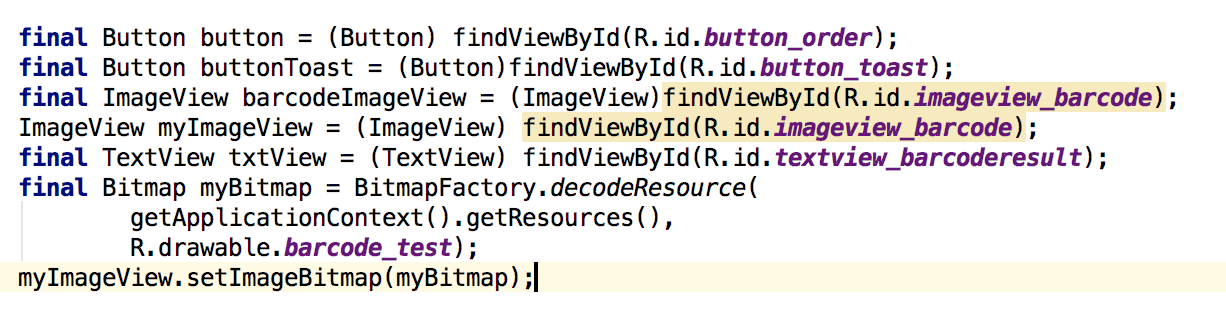


这些类都可以设置监听，接下来让我们在app中实现一个长按识别二维码：

1. 二维码图片放入文件夹app/src/main/res/drawable
2. 在activity\_main.xml文件中的注释加入*<!--android:src="@drawable/barcode\_test"-->* ImageView的</>中。
3. 将下面的代码加入到你刚才的Button对象下面（如图）：

文件step2

**final** ImageView barcodeImageView = (ImageView)findViewById(R.id.***imageview\_barcode***);  
ImageView myImageView = (ImageView) findViewById(R.id.***imageview\_barcode***);  
**final** TextView txtView = (TextView) findViewById(R.id.***textview\_barcoderesult***);  
**final** Bitmap myBitmap = BitmapFactory.*decodeResource*(  
 getApplicationContext().getResources(),  
 R.drawable.***barcode\_test***);  
myImageView.setImageBitmap(myBitmap);



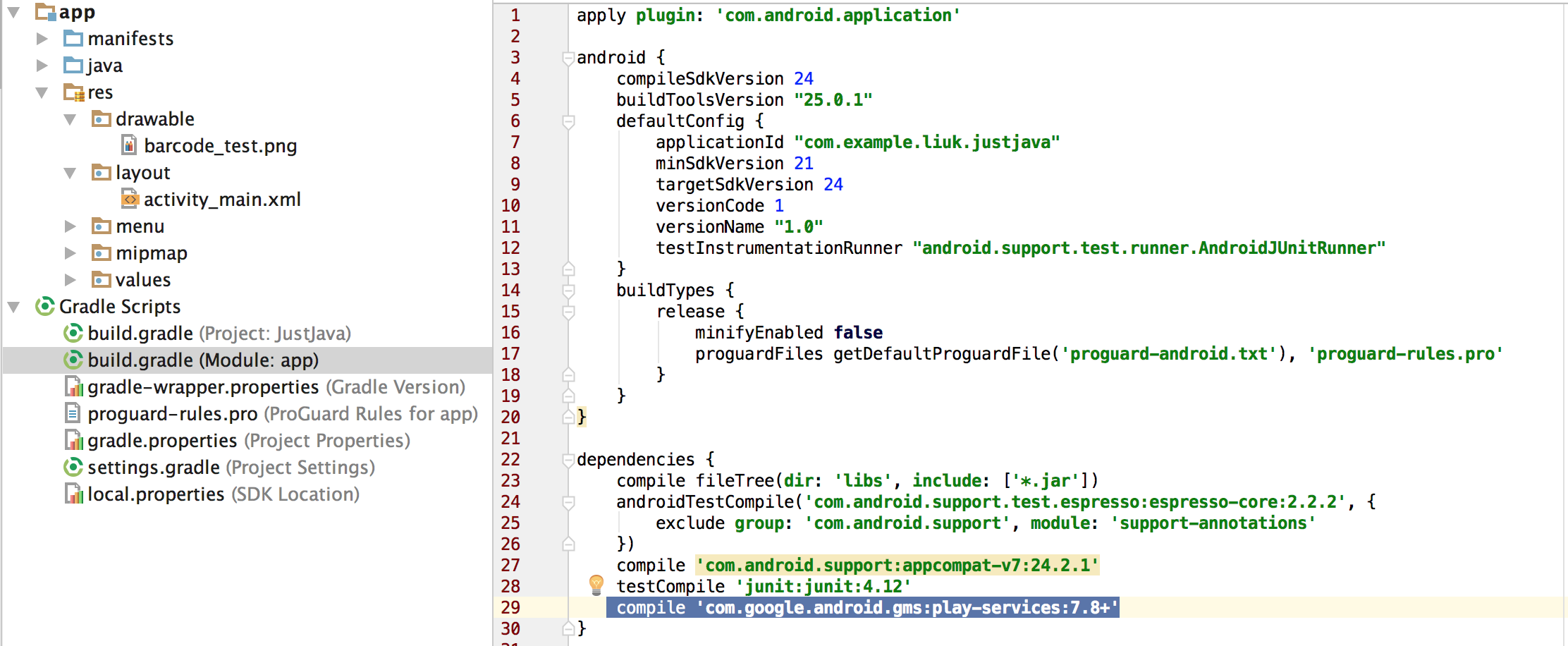
1. 将以下代码放在onCreate方法里面：

文件step3.txt

barcodeImageView.setOnLongClickListener(**new** View.OnLongClickListener(){  
 @Override  
 **public boolean** onLongClick(View v) {  
 Toast.*makeText*(MainActivity.**this**,**"正在识别二维码"**, Toast.***LENGTH\_SHORT***).show();  
 BarcodeDetector detector =  
 **new** BarcodeDetector.Builder(getApplicationContext())  
 .setBarcodeFormats(Barcode.***DATA\_MATRIX*** | Barcode.***QR\_CODE***)  
 .build();  
 **if**(!detector.isOperational()){  
 txtView.setText(**"Could not set up the detector!"**);  
 **return true**;  
 }  
 Frame frame = **new** Frame.Builder().setBitmap(myBitmap).build();  
 SparseArray<Barcode> barcodes = detector.detect(frame);  
 Barcode thisCode = barcodes.valueAt(0);  
 txtView.setText(thisCode.**rawValue**);  
 **return false**;  
 }  
});

这个时候出现问题了，不要慌，在下面这个文件里加入：

compile **'com.google.android.gms:play-services:7.8+'**

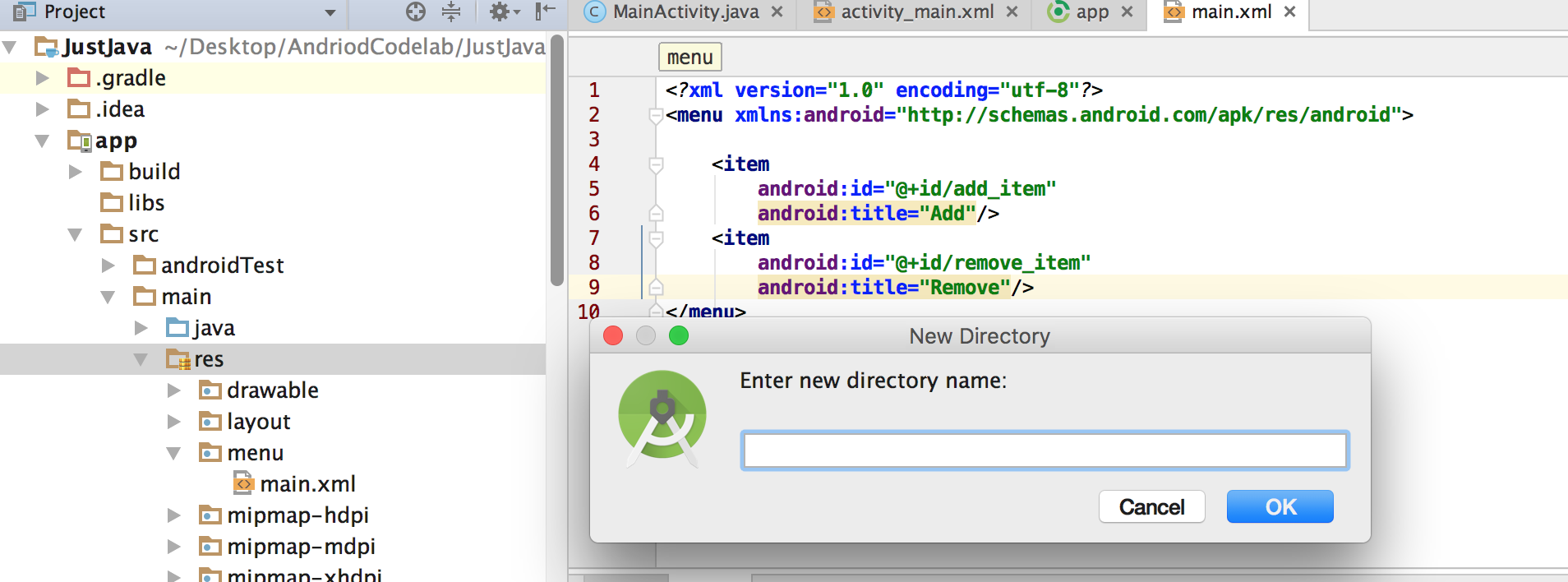
****

并且确定你的Google Play Services版本在26以上，升级步骤如下。

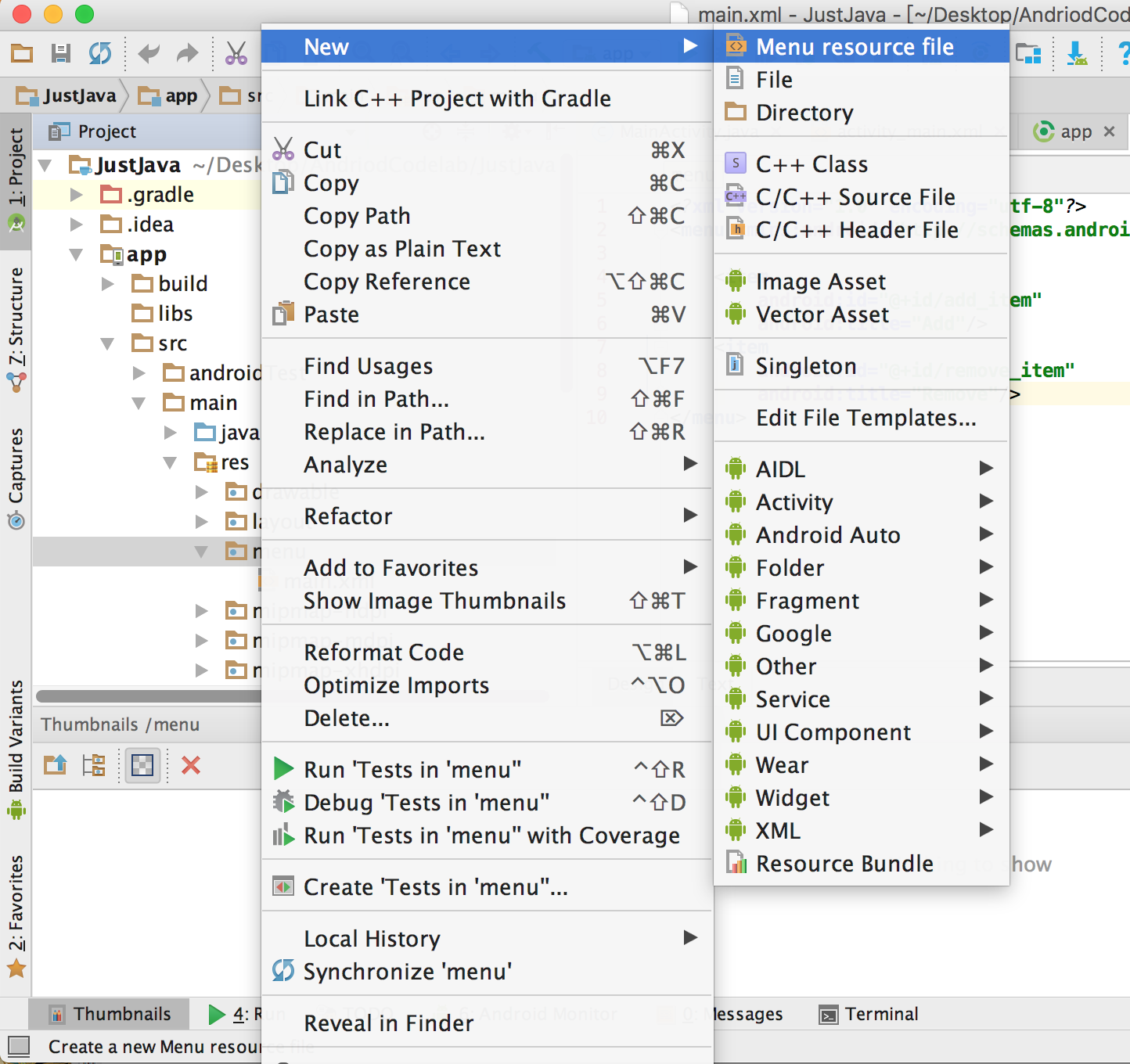
|  |  |
| --- | --- |
| Macintosh HD:Users:liuk:Downloads:32ba4ede750d7a39.png | Macintosh HD:Users:liuk:Downloads:4a55603b66313bba.png |

当问题解决，你的app就实现里以上的功能。

最后，让我们实现菜单功能：在app/src/main/res上右键new->Directory名叫menu。



右键menu new->menu resource file,取名main。



在main.xml中加入以下语句：

*<?***xml version="1.0" encoding="utf-8"***?>*<**menu xmlns:android="http://schemas.android.com/apk/res/android"**>  
  
 <**item  
 android:id="@+id/add\_item"  
 android:title="Add"**/>  
 <**item  
 android:id="@+id/remove\_item"  
 android:title="Remove"**/>  
</**menu**>

再在MianActivity中重写以下方法。

文件step4.txt

@Override  
**public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***main***,menu);  
 **return true**;  
}  
  
@Override  
**public boolean** onOptionsItemSelected(MenuItem item) {  
 **if**(item.getItemId()==R.id.***add\_item***){  
 Toast.*makeText*(MainActivity.**this**,**"Add success"**,Toast.***LENGTH\_LONG***).show();  
 }**else if**(item.getItemId()==R.id.***remove\_item***){  
 ((TextView)findViewById(R.id.***textview\_barcoderesult***)).setText(**"NULL"**);  
 Toast.*makeText*(MainActivity.**this**,**"Remove success"**,Toast.***LENGTH\_LONG***).show();  
 }  
 **return true**;  
}

这时再去运行app，就会有右上角的菜单栏里